Trust management in the public-key infrastructure

Tim Moses

Abstract

Public-key infrastructure manages trust in exchanges conducted by email, over the web and by other electronic means. The principal elements used for maintaining that trust are the contents of the certificates and the security safeguards in effect in the environments of the various parties involved. These two elements are derived by a risk management procedure from the business purpose of the exchanges, as captured in the certificate policy. In this paper we describe a high-level procedure for deriving certificate contents and security safeguards from the business purpose associated with the keys, by way of a certificate policy and security policies for each of the subscriber, relying party and authority environments.

Introduction

Trust relationships

Before embarking upon a discussion of trust management in the public-key infrastructure, we need to find a useful definition for the word "trust". [X.509] defines trust in this way:

> *"Generally, an entity can be said to "trust" a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects."*

Naturally, the first entity makes this assumption only about a relevant area of the second entity's behaviour, and so the trust between them is limited only to that area. A breach of trust is different from a simple ***violation*** (a traffic offence, for instance, is not a breach of trust). Rather, the essence of trust lies in the disappointment of a trusting party's reasonable expectation of someone who was under a higher than normal duty to fulfill those expectations.

In this discussion we are concerned with behaviour related to the distribution and use of public keys for electronic commerce. Different types of trust relationship are capable of conveying different types of assurance between the

parties. A trust relationship based upon public-key technology is intended to ensure the authenticity of the second entity's identifying descriptor and the enforceability of commitments undertaken by both entities.

Conventional trust relationships

Trust is a well-established concept, and there are many examples of conventional trust relationships, including those between a bank and its account holders, between an employer and its employees, between a government and its citizens, between the media and its subscribers, between an industry association and its members and so on. We will see later how existing conventional trust relationships play an essential role in establishing new trust relationships based on public-key technology.

Public-key-based trust relationships

In the realm of public-key technology, a necessary step towards establishing a trust relationship is for the first entity to import a public key from the second one and protect its integrity for storage or communication to other entities. The entity that imports the public key is known as the relying party, because it intends to ***rely*** upon the public key for protecting subsequent exchanges with the key-holder (the entity from whom the key is imported). This entity relationship is shown in Figure 1.

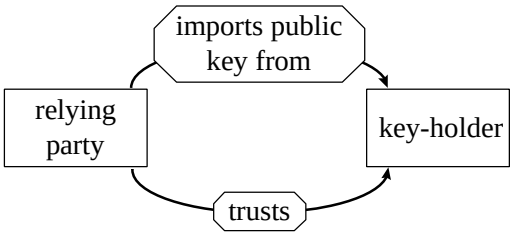

Figure 1 - Entity relationship

Any entity may act, simultaneously, as both a relying party and a key-holder. But, for the sake of simplicity, we will separate these two roles throughout this discussion.

In order to avoid confusion between the two parties, the public key import operation must be performed in a manner that preserves the key's authenticity and integrity (i.e. it must be received, unmodified, from the correct key-holder) and its "clarity" (i.e. the relying party's understanding of the approved uses for the public key must be the same as the key-holder's understanding). These security properties can only be established by means of an existing trust relationship capable of conveying the necessary assurances. So, it appears to be axiomatic that a trust relationship cannot be "created" where there is no existing trust relationship. Rather, existing trust relationships can only be "qualified" and "combined" to form trust relationships with new

characteristics. So, we will concern ourselves here with ways of building trust relationships with desirable characteristics, based on public-key technology, and using existing conventional trust relationships as their starting point.

One way of building on an existing trust relationship, to form a new trust relationship based on public-key techniques with integrity and clarity, is illustrated by the data-flow diagram of Figure 2.

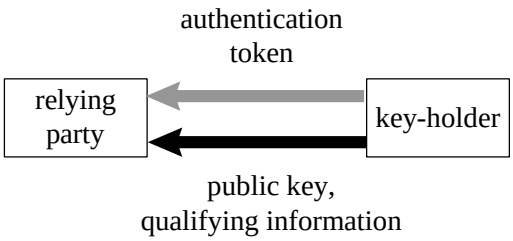

Figure 2 - Data flow

In the diagram, the lighter arrow represents an exchange in an existing trust relationship. The darker arrow represents an automated exchange whose trustworthiness depends upon the information conveyed in the first exchange.

In the example, the authentication token may take the form of a displayable string of characters (which the relying party can conveniently read and enter at a computer keyboard) or some pre-existing shared secret information, which is linked with the public key and qualifying information. Proper transfer of the authentication token relies upon the existing trust relationship, and the authenticity and integrity of the public key and its qualifying information can then be protected by relying on this authentication token.

An essential component of the qualifying information is an identifying descriptor for the key-holder. The descriptor may be unique or shared, or some combination thereof. Sometimes it is the key-holder's name, but this is not necessarily the case. In many applications, the relying party's end-goal is to associate a privilege with the key-holder, and it will use the public key to authenticate the key-holder merely as an initial step in controlling the granting of that privilege. In other circumstances, the qualifying information may indicate directly that the key-holder possesses the required privilege. In self-service and inter-personal messaging applications, the key-holder's identifying descriptor is commonly sufficient.

The property of clarity may be implemented by the qualifying information in a number of different ways. It may be partially and implicitly expressed in the type of the public key, because for technical reasons not all public keys can be used for all business purposes. It may be explicitly encoded in key-usage codes and it may be included by reference in the form of certificate policy identifiers.

Trust and risk

According to the X.509 definition of trust, the risk that the key-holder might fail to behave as expected naturally attaches to the relying party. Some examples of the elements of risk in a public-key-based trust relationship are:

- the identifying descriptors associated with a key are incorrect or misleading;
- the public-key holder's private key has been discovered by another entity;
- the public-key holder's implicit privilege has been withdrawn recently;
- the public-key holder has a prevailing right not to be bound by its signature in the way the relying party expects;
- the public-key holder does not adequately protect the confidentiality of the sensitive information that it is entrusted with;
- etc..

For dealings between individuals, where the relying party has a close and long-standing relationship with the community of key-holders, this allocation of risk is appropriate, because the relying party is able to evaluate its risk and decide whether or not to accept it. But, in electronic-commerce, the relying party may either be unqualified to evaluate its risk or will evaluate it and choose not to accept it. In this case it will attempt to rely upon external sources of trust to shield it from risks that it cannot cost-effectively eliminate and which can make an injured party whole should a failure occur.

Risk management may include a number of strategies:

- ***Minimize*** (i.e. reduce the probability that a loss-causing event will occur). To minimize risk, the risk taker attempts to reduce the probability of a loss-causing event as much as practicable. A loss is any economically significant failure of the key-holder's or relying party's legally enforceable expectations. In other words, to avoid a loss-causing event, each party attempts to perform according to the other's expectations.
- ***Avoid or contain consequences***. If a failure of one party's expectations occurs, the other party tries to reduce the economically significant consequences, as much as is practicable. For example, disclosure of the key-holder's private key may breach an obligation to a relying party, but if the key-holder informs the relying party before the relying party suffers any harm, then the damages due to the breach are only nominal.

- ***Bear the residual risk***. It generally isn't possible or cost-effective to reduce the risk of a loss-causing event to zero, so the residual risk must be borne, usually by spreading it among a large group of risk bearers, so that financially the risk amounts to a fixed, budgetable expense rather than the unpredictable possibility of a crippling loss. This spreading can often happen over time, so that the time value of money is involved, as well as over geography, societies, etc..

Suitable safeguards are chosen according to the intended business purpose of the key by means of the risk management process, as shown in Figure 3.

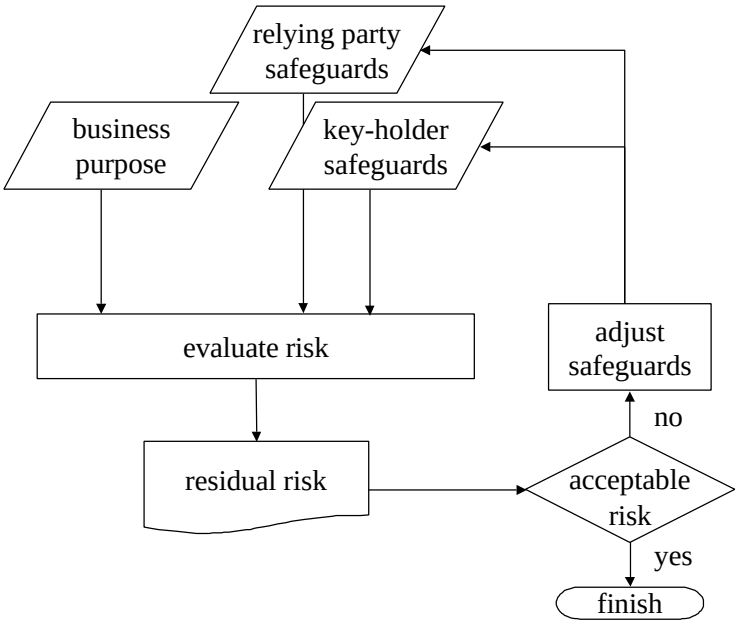

Figure 3 - Safeguard selection

The safeguards applied by the relying party and the key-holder are evaluated in the context of the intended business purpose of the public key, and if the residual risk is not acceptable, then adjustments to the choice of safeguards must be made.

It is not our intention to suggest that safeguard selection can proceed literally in such a procedural manner; it is still necessary to apply a great deal of judgment and experience to the process.

Independent of the question of selecting suitable safeguards, there is the question of "assurance". That is, if, when properly implemented, the chosen safeguards reduce the risk to an acceptable level for the intended business purpose, each party requires assurance that the safeguards are indeed properly implemented, both in its own environment and in those of the other parties.

In the majority of its dealings, the relying party will be able to identify a suitable trusted third party in the form of an appropriate traditional source of trust to assist it in bearing its risk, suitable sources include, but are not limited to, an employer, bank, doctor, government, etc..

The trusted third party

The involvement of a trusted third party, or "authority", is shown in Figure 4.

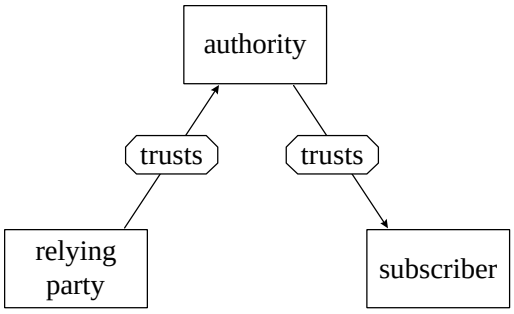

Figure 4 - Trusted third party

When reliance is placed on an authority, the key-holder is commonly referred to as a subscriber, because, sometimes, the key-holder elects to subscribe to a service operated by the authority. As we place greater and greater reliance in electronic commerce systems, the authority may be required to accept a significant measure of risk from the relying party. An authority with a close and long-standing relationship with the subscriber community will be better placed to mitigate risk associated with registering and controlling the behaviour of that community. In order to restrict its risk acceptance to matters that are under its ***direct*** control, the authority will have to redistribute that portion of its risk that is associated with other matters to the parties, such as the subscriber and relying party, who ***do*** have direct control over them. Suitable mechanisms for redistributing risk are discussed further, below.

It is often stated that trust is not transitive. However, the situation shown in Figure 4 is a common situation in which the trust is, clearly, transitive: the relying party trusts the authority; the authority trusts the subscriber; so, the relying party trusts the subscriber. It is the case, however, that trust suffers a dilution if chains of trust are overly extended.

Note that the main motivation for introducing the authority in Figure 4 is described in terms of risk management, not in terms of the technical challenges

associated with large-scale systems. Each exchange deposits a quantum of risk with a party, so the aggregate risk grows in proportion to the number of exchanges. And if each member of the community engages in exchanges at the same rate, then the aggregate risk grows in proportion to the size of community. Therefore, risk increases with the size of community. And, risk management generally imposes a more severe limit on scale than purely technical considerations do.

In general, a single relying party may rely on more than one primary source of trust for its dealings in different aspects of its life. But, for the sake of clarity, we will initially consider just one such reliance relationship.

Redistribution of risk

Four types of mechanism are available for transferring a portion of the relying party's risk to the subscriber or authority:

- existing public law (such as statutes, regulations, case law and other governmentally imposed rules);
- digital signature law;
- direct control; and
- contract.

Existing public law includes no provisions specific to the allocation of risk amongst participants in a public-key infrastructure. It does, however, contain provisions for consumer protection, fraud, deception, etc..

Digital signature law in effect within a particular jurisdiction may prescribe how risk shall be apportioned between relying party, subscriber and authority. However, there is currently no universally-applicable framework that encompasses the activities of the parties, and there is no certainty that such a framework will ever exist. Between those jurisdictions in which legislation has been enacted, there is little consistency of approach. Furthermore, since relying parties, subscribers and authorities involved in a particular transaction may be subject to the law in different jurisdictions, predicting which law will apply from the choices available is difficult.

Direct control involves the enforcement of risk mitigating measures by one party on another. This enforcement takes the form of imposed operating procedures which are maintained by means of audit and the direct assumption of certain responsibilities. This approach is most effective within the bounds of a single legal entity, because organizations doing business "at arms-length" do not generally allow one another the necessary degree of access.

Contract is required to clarify the allocation of responsibilities in all other situations. However, unless standard clauses are used, contracts can be costly to draft, negotiate and close.

The most significant feature of the legal and regulatory framework is that participants cannot effectively claim ignorance of its provisions. Naturally, direct control and contract cannot override the provisions of the applicable public law within a jurisdiction, unless the public law permits it. But, unless and until a uniform global legal framework is established, contract must be the foundation for risk management *between arms-length entities* and direct control can be effective *within the boundary of a single legal entity*.

Compound trust relationships

Relying parties tend to redistribute risk to authorities which are "close to", or have a direct and long-standing relationship with, the subscriber community. The main reason for this is that proximity facilitates familiarity, so people close at hand have access to better information and evidence. In economic terms, they can confirm the accuracy of certified information more cheaply and easily than remote persons, whose information is more likely to be derivative, based on heuristic assumptions, etc..

Such authorities may take one of two forms:

- an authentication authority; or
- a certification authority.

According to our definition, an authority is an ***authentication*** authority if it has only conventional trust relationships with the relying party and the subscriber, and the public-key relationship exists directly only between the relying party and the subscriber. On the other hand, an authority is a ***certification*** authority if a public-key relationship is established between the authority and the relying party and between the authority and the subscriber as a precursor to the establishment of the direct public-key relationship between the relying party and the subscriber. Later, we will see how to build practical compound trust structures containing either or both types of authority.

Figure 5 shows the trust relationships operating between a relying party, a subscriber and an ***authentication*** authority. In a registration process (1), the subscriber provides an authentication token to the authentication authority. Upon successful registration, the authority makes the authentication token, and applicable qualifying information, available to the relying party (2). A relying party can then obtain the subscriber's public key directly from the subscriber and use the authentication token to confirm its authenticity and suitability to its business purpose (3).

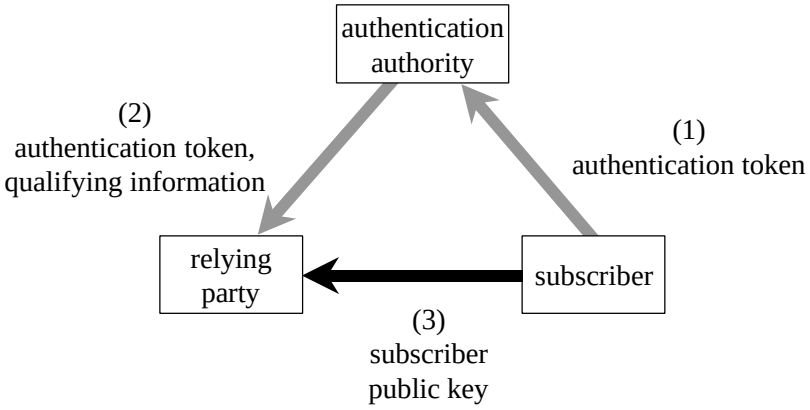

Figure 5 - Authentication authority trust relationships

Figure 6 shows the trust relationships operating between a relying party, a subscriber and a ***certification*** authority.

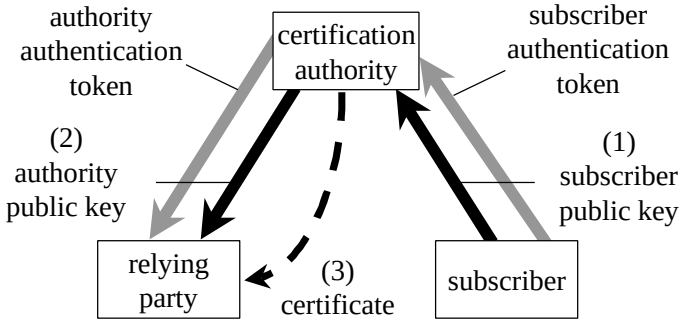

Figure 6 - Certification authority trust relationships

In this case, the subscriber public key (1) is supplied to the certification authority by means of an authentic protocol and (likewise) the authority's public key is supplied to the relying party by means of an authentic protocol (2). Subsequently, the subscriber public key and qualifying information are supplied to the relying party either directly from the authority, or by some other communications path, with its authenticity, integrity and clarity protected by a digital signature applied by the authority (3). The corresponding data structure is called a certificate and the most common protocol for implementing this scheme is [X.509]. The certificate can be viewed as the secure protocol by which the certification authority communicates trust to the relying party.

The main advantages of a certification authority over an authentication authority are:

1. evidence of the role of the certification authority appears in the sequence of certificates used by the relying party to validate the subscriber's public key, whereas evidence of the role of the authentication authority does not;

2. consequently, in the case of the certification authority, the relying party identifies the basis of its trust with the authority that introduced it to the subscriber, rather than with the subscriber itself, as is the case with the authentication authority;

3. the certification authority can automatically revoke the trust in the subscriber, whereas the authentication authority cannot; and

4. standard protocols are defined for the function of the certification authority, but not for the function of the authentication authority.

The main advantages of an authentication authority over a certification authority are:

1. the authentication authority does not have to be implemented in an automated information processing system, whereas the certification authority does; and

2. when using an authentication authority, the certificate path contains one fewer certificate than it does when using a certification authority.

More complex trust structures

Figure 7 illustrates the two elementary trust transformations, based upon the authentication authority and the certification authority, that were introduced above. The diagrammatic conventions in this diagram are identical to those in the earlier diagrams. The light arrows represent exchanges in a conventional trust relationship. The dark arrows represent exchanges in a trust relationship derived from the conventional trust relationship using the first exchange. And the broken arrows represent certificates.

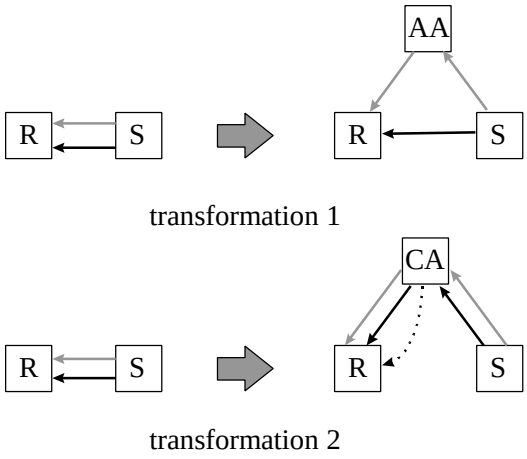

Figure 7 - Elementary trust transformations

These transformations can be applied repeatedly to form more complex compound trust models. Five such models of particular interest are shown in Figure 8 to Figure 12. The characteristics of these models are discussed further below.

Subscriber registration authority

The subscriber registration authority model is shown in Figure 8.

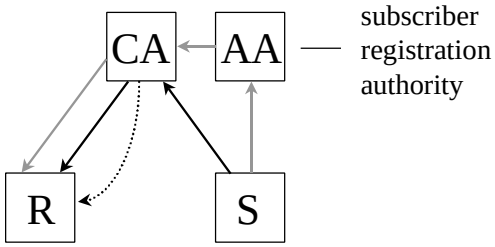

Figure 8 - Subscriber authentication authority

This model results from applying *transformation 2* and then *transformation 1* to the subscriber relationship. It is useful when the CA is remote from the subscriber community. In this configuration, the authentication authority is commonly referred to as a subscriber registration authority. Although there are two authorities, there is only one certificate, and the involvement of the authentication authority is invisible to the relying party.

Direct cross-certification

The direct cross-certification model is shown in Figure 9.

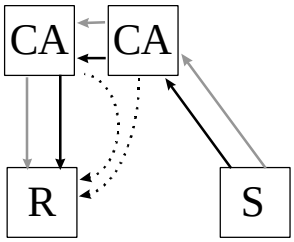

Figure 9 - Direct cross-certification

This model results from applying *transformation 2* twice to the relying party relationship. Direct cross-certification is an applicable model when authorities operated by separate organizational entities enter into a direct trust relationship. In this case, there are two authorities and two certificates, and the involvement of each authority is visible to the relying party.

Two-tier hierarchy

The two-tier hierarchy model is shown in Figure 10.

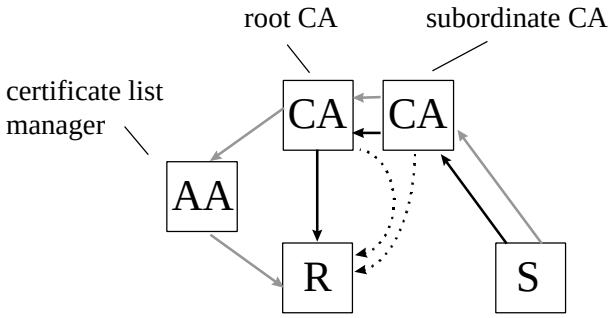

Figure 10 - Two-tier hierarchy

This model results from applying *transformation 1* to the relying party relationship shown in Figure 9. In this case, the authentication authority is more commonly referred to as a certificate list manager. The two-tier hierarchy is an applicable model when the certificate list manager and the subordinate CA are operated by separate organizational entities and their trust relationship is facilitated by a third entity, which operates the root CA. There are three authorities, but only two certificates and the involvement of the authentication authority is not recorded in the list of certificates, which, in conjunction with the business transaction, form the complete evidence.

Hub certification authority

The hub certification authority model is shown in Figure 11.

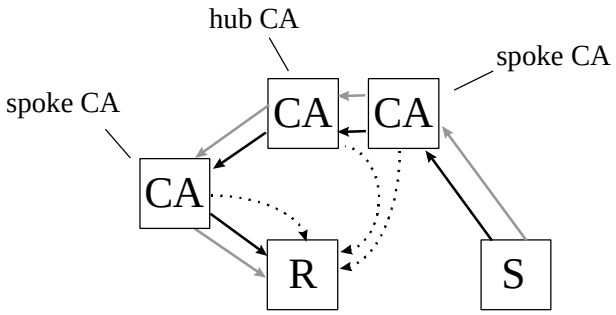

Figure 11 - Hub certification authority

This model results from applying *transformation 2* to the relying party relationship shown in Figure 9. The hub certification authority is an applicable model when the two spoke certification authorities are operated by separate organizational entities and their trust relationship is facilitated by a third entity, which operates the hub CA. There are three authorities and three certificates, so the role of each authority is recorded in the list of certificates that form the evidence. (The hub CA is sometimes called a bridge CA.)

Hub authentication authority

The hub authentication authority model is shown in Figure 12.

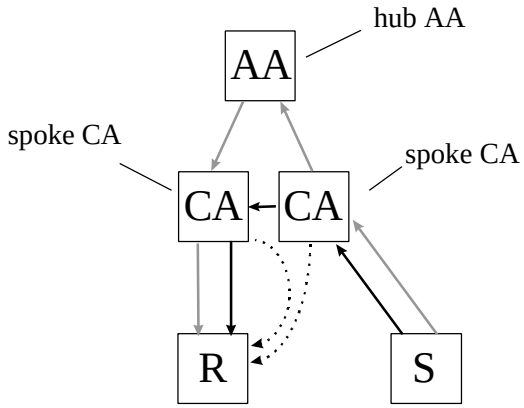

Figure 12 - Hub authentication authority

This model results from applying *transformation 1* to the relationship between the two authorities shown in Figure 9. The hub authentication authority is an applicable model when the two spoke certification authorities are operated by separate organizational entities and their trust relationship is facilitated by a third entity, which operates the hub authentication authority. This model is useful when the third entity is not equipped to operate an automated information system. There are three authorities, but only two certificates. Therefore, the involvement of the hub authentication authority is not recorded in the list of certificates that form the evidence.

Summary

The trust transformations described above may be repeatedly applied to create ever more elaborate trust models. However, the five shown above are the ones of most practical interest.

Different trust models are suited to different business situations. But, no matter which compound trust model is chosen, the relying party expectation is that its trust requirements are satisfied by the authority upon which it relies directly, and if that authority makes private arrangements to redistribute its risk to other authorities or subscribers, then this in no way diminishes its obligation to its relying parties. So, it must take whatever measures are necessary to control the behaviour of other authorities and subscribers whose keys it has certified, directly or indirectly, so that its risk remains under control.

In practical terms, the trust brand displayed to the relying party will be that of the ***certification*** authority upon which it relies directly. Where an authentication authority is involved, its role is invisible to the relying party at the time of validating the subscriber certificate. Although a significant measure of risk may be accepted by an authentication authority, the relying party appears to rely solely on the certification authority whose public key it has imported directly.

Certificate and security policy

The effectiveness of risk redistribution from an authority or relying party to a another authority or subscriber is determined by two things: the certificate policy agreed between the parties and the security policy under which each party operates. Let's clarify what we mean by these two terms.

Certificate policy specifies the business purpose for which the public key is approved.

Security policy is a statement of requirements for the safeguards that are to be applied in an entity's environment.

Relationship between certificate policy and security policy

Certificate policy is related to the business purpose of the public key, and security policy is related to security safeguards. As illustrated in Figure 3, business purposes and safeguards are related by the risk management process. Figure 13 shows the relationship between certificate policy and security policy. It is similar to Figure 3 but now we have introduced a third entity: the authority. Separate security policies are required for each type of entity in the PKI: the relying party, the subscriber and the authority.

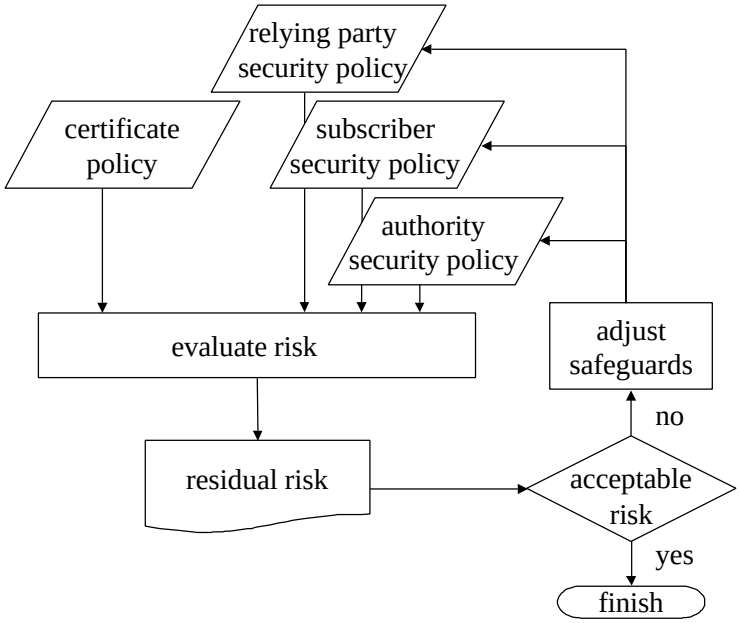

Figure 13 - Relationship between security policy and certificate policy

Policy development proceeds generally by the following steps.

1. A policy authority, which impartially represents the participants in the public-key infrastructure, defines the certificate policy.

2. The policy authority performs a risk assessment, based on the certificate policy, and chooses security policies for the relying party, subscriber and authority environments.

3. The policy authority operates (or appoints one or more service providers to operate) certification services in conformance with the authority security policy. If it chooses to appoint service providers, then it will, more likely, select one of the small number of authority security policies implemented by the service provider. These may have been developed without regard to the intended purpose of the subscriber public keys that they certify.

4. The policy authority may prescribe the security policy to be enforced in each of the relying party and subscriber environments. Alternatively, the relying party and subscriber may determine their own security policies, and commit that their certificates are managed in a way that is consistent with the certificate policy.

The policy authority role is usually performed by one of the traditional sources of trust mentioned above. Alternatively, participants may agree upon

certificate policy bilaterally, without the intervention of an independent policy authority.

Generally, security policy is a great deal more detailed than certificate policy.

Assurance

As mentioned earlier, in addition to the question of selecting suitable safeguards for the intended purpose, it is necessary to ensure that the safeguards are applied effectively. This is generally achieved by an independent audit process.

If the policy authority performs the risk assessment, and imposes security policy upon the participants, then the audit is concerned with adherence to the security policy. If, on the other hand, the policy authority merely defines the certificate policy, then the audit is concerned with the ability of the participants to meet their obligations under the certificate policy.

Multiple policy regimes

Public-key infrastructure entities have to participate in multiple policy regimes. For instance, a manufacturing company may participate in public-key infrastructures operated by the relevant tax authority, its supplier community, its vendor community, its banking and insurance communities, etc.. Each of these public-key infrastructures will develop independently and so will operate under different certificate policies. Therefore, communities of subscribers need to have their certificates accepted by more than one community of relying parties.

Because of the amount of prescriptive detail in a ***security*** policy, the probability that an entity can act simultaneously in conformance with multiple security policies is low. The best hope for addressing this is to implement them supremum set of safeguards across all the quantifiable elements of the required policies.

If communities adopt common ***certificate*** policies and the corresponding ***security*** policies are derived by risk management, then it will be possible for entities operating in conformance with one ***security*** policy to act simultaneously in conformance with multiple ***certificate*** polices.

The authority, subscriber and relying party behaviour may have to be modified slightly in accordance with the security policy provisions. For instance, the degree of protection afforded to the subscriber private key, including the authentication requirements for activating the private key, may vary. These factors are prescribed by the ***security*** policy, not by the ***certificate*** policy. The risk-bearing entity should derive the security policy from the certificate policy

and be able to enforce all technical aspects of its security policy by reliable technical means.

Model certificate policy

This section describes a model for certificate policy. In the presentation of this model, the subject is either an end-user key-holder or a certification authority issuing subscriber certificates or cross-certificates. The issuer is either an end-user relying party or a certification authority issuing cross-certificates. In a later section we will describe how elements of this policy model should affect the contents of a certificate issued by one authority to another authority in a compound trust structure.

A conformant certificate policy should contain the following elements.

Issuer – The name or qualification of the relying party or issuing CA.

Subject – The name or qualification of the subscriber or subject CA. If a name, then the issuing authority must ensure that the name is unique within its own name-space, or define a mapping between the name and a name that is unique within its own name-space.

Trust mark (Optional) – The visual trust mark by which the subject authority is recognized. If the relying party is an end-user, then this mark should be displayed as a visual reminder that it is relying upon this authority in accordance with the policy.

Community (Optional) – A set defining the relationship between the subject authority and its direct subscribers. Example values include "employee", "account holder", "customer", "patient", "citizen", "unaffiliated", "authority", etc..

Hierarchy level (Optional) – The hierarchical level of the subject CA. Required to be present if **Community** includes the value "authority". If the subject authority issues only end-user certificates, indicated by the absence of the "authority" value in the **Community** element, then its hierarchical level is zero. If the subject authority issues certificates to other certification authorities, indicated by the presence of the "authority" value in the **Community** element, then its hierarchical level is one greater than the level of its subject authority that has the highest hierarchical level.

Names (Optional) – If the subject is a certification authority, then this parameter indicates the name-space sub-trees in which it issues certificates.

Key usage – If the subject is an end-user, then this parameter indicates the approved uses to which its public key may be put. If the subject is a certification authority, then this parameter indicates the approved uses to which

the certificates it issues may be put. Example values include “data encryption key transport”, “digital signature” and “commitment”. Note that, in this case, the subject CA’s own key usage certificate extension will be “certificate signing” and or “CRL signing”.

Policy (Optional) – May be present if the subject is a certification authority. The identifier and human-readable name allocated by the subject authority to the policy.

Limits (Optional) – Per-transaction and aggregate liability limits. Only required to be present if the value of **Key usage** includes “commitment”.

Non-disclosure (Optional) – A statement on non-disclosure of information provided in encrypted form by relying parties to the subject community, including decryption key back-up provisions, where appropriate. Should be present if the value of **Key usage** includes “data encryption key transport”.

Relying party notice – Text to be displayed to the relying party at the time of reliance.

Certificate access method (Optional) – The method by which certificates can be obtained from the subject authority’s certificate repository. Access to the certificate repository may be restricted to qualified relying parties. May be present if the subject is a certification authority.

Revocation access method (Optional) – The method by which revocation information can be obtained from the subject authority. Access to the revocation information may be restricted to qualified relying parties. Should be present if the subject authority publishes revocation information.

Notice registration method (Optional) – If the subject is a certification authority, then the method by which a relying party should register to receive notices issued by the authority. Should be present if the subject authority expects to issue notices.

Enquiry method (Optional) – The method by which the relying party should submit enquiries. Should be present if the subject authority can respond to enquiries.

Jurisdiction (Optional) – If the subject is a certification authority, then the legal jurisdiction in which it operates. Should be present if the value of **Key usage** includes “commitment”. May be present if the value of **Key usage** does not include “commitment”.

Arbitration (Optional) – If the subject is a certification authority, then the arbitration body recognized by it. Should be present if the value of **Key usage**

includes “commitment”. May be present if the value of **Key usage** does not include “commitment”.

Policy authority (Optional) – The name of the authority which administers the certificate policy.

Notice method (Optional) – The method by which notices should be sent by the subject authority to the relying party.

Start date – The time and date before which the relying party does not qualify as a relying party.

End date – The time and date after which the relying party does not qualify as a relying party.

References (Optional) – If the policy authority prescribes elements of security policy in one or more of the operating environments, then these shall be referenced here.

Certificate contents

In this section we describe the contents of the X.509 v3 cross-certificate extensions that are required to enforce business controls required by a conformant policy, when the certificate policy is agreed between two authorities.

Basic constraints - If the “authority” value is not present in the certificate policy **Community** element, then the basic constraints extension should identify that the subject is an authority and the corresponding path length constraints value should be set to zero, indicating that it should issue certificates only to end-users. If the “authority” value is present in the certificate policy **Community** element, then the **Hierarchy level** value should be used as the path length constraints value in the cross-certificate. This extension should be marked critical.

Name constraints - The value of **Names** from the certificate policy. This extension should be marked critical.

Certificate policy - The cross-certificate should contain a certificate policies extension which includes the **policy identifier** from the certificate policy. The human-readable policy name may be included as a policy qualifier. May be marked non-critical.

Policy constraints – The cross-certificate should include a policy constraints extensions which allows policy mapping and requires that policy identifiers be present in subsequent certificates. This extension should be marked critical.

Policy mappings - Identifies the issuer's certificate policy equivalent to the policy described in the certificate policy. Policy mappings are only used between identical certificate policies that have been assigned different identifiers. May be marked non-critical.

Key usage – Certificate signing and CRL signing. Should be marked critical.

Model security policy

Elements of security policy apply to each of the parties in the trust structure. Root certification authorities need to ensure that their subordinate CAs and subscribers adhere to those elements of security policy that apply to them. Otherwise, their risk is not properly under control. Similarly, hub certification authorities need to ensure that their spoke CAs and subscribers adhere to those elements of security policy that apply to them.

A representative selection of technical elements of security policy for each of the environments is given below.

Authority

- Whether private key operations must be performed in a tamper-resistant enclosure.
- The requirements for authenticating an applicant's identity and privilege during enrollment.
- Whether and how revocation information must be issued, and the maximum latency.
- Authentication requirements for the approval of certificate issuance.

Subscriber

- Authentication requirements for activating the private key.
- The set of acceptable authentication methods for activating the private key.
- Whether private key operations must be performed in a tamper-resistant enclosure.
- Whether the decryption private key may be released for emergency back-up purposes.
- Whether user confirmation is required for each individual use of the signature private key.

Relying party

- The authority public key in which direct reliance must be placed.
- The set of acceptable integrity mechanisms for protecting the authority public key.
- Minimum entropy requirements for the integrity algorithm key.
- Whether revocation information must be checked.
- The initial policy set for the X.509 certificate validation procedure.
- Whether a notice must be displayed to the relying party.
- Whether transactions must be time-stamped.
- Whether valid transactions must be archived.

Both technical and non-technical elements of security policy applicable to public-key systems are exhaustively enumerated in [PKIX].

Summary

We have seen how certificate contents and security safeguards can be derived, by a risk management process, from a statement of the intended purpose of the public key in the form of a certificate policy, which conforms with a model certificate policy described here.

Practical trust models

The technical requirements of interoperability between autonomous trust domains can be achieved in any one of four main ways:

- Direct cross-certification;
- Two-tier hierarchy;
- Hub certification authority; and
- Hub authentication authority.

There follows some discussion of each alternative. For the sake of clarity, just two arms-length organizations are shown in a bilateral relationship and in a multiple-policy environment. In practice, however, the number of inter-operating organizations will be greater than two and some relationships may be unilateral, but these complications do not alter the conclusions. The diagrammatic convention in the diagrams is that thin lines represent unqualified trust (i.e. a quality of trust that is appropriate to all the relevant certificate

policies). Thick lines represent “qualified” trust (i.e. trust in conformance with an identifiable certificate policy).

Direct cross-certification

The trust relationships in the direct cross-certification trust model are shown in Figure 14. One authority, acting as an agent for its community of relying parties, evaluates the risk of accepting certificates issued by the other authority, and places the appropriate controls in the cross-certificates that it issues. Relying parties ensure that keys are used only for appropriate business purposes by specifying the initial policy set for certificate path validation. The cross-certificate, issued by one authority to the other, identifies all the business purposes for which the subject authority is considered acceptable.

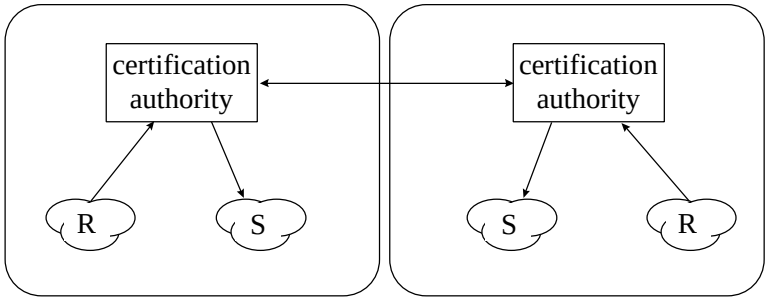

Figure 14 - Direct cross-certification

The main advantage of this approach is that it can function when no suitable third party is available to facilitate the formation of the relationship.

The main drawback of this approach to inter-domain trust management is that the cost of the risk assessment associated with entering into the trust relationship is prohibitive when the number of relationships is more than a small number. In order to ameliorate this drawback, we can either reduce the cost per relationship, or reduce the number of relationships. Methods of reducing the number of relationships are discussed below.

Two-tier hierarchy

The trust relationships in the two-tier hierarchy are shown in Figure 15. Relying parties throughout the domain of the root certification authorities import the public keys of those root CAs under the control of their local certificate list manager authentication authority. The root CAs issue certificates in accordance with different policies to subordinate certification authorities, which, in turn, issue end-user certificates to subscribers. Relying parties ensure that keys are used only for appropriate business purposes by forming certificate chains that emanate from the authority which operates the applicable certificate policy. The security policy in effect in the subscriber

environment must be consistent with all the certificate policies operated by all the root CAs to which it subscribes.

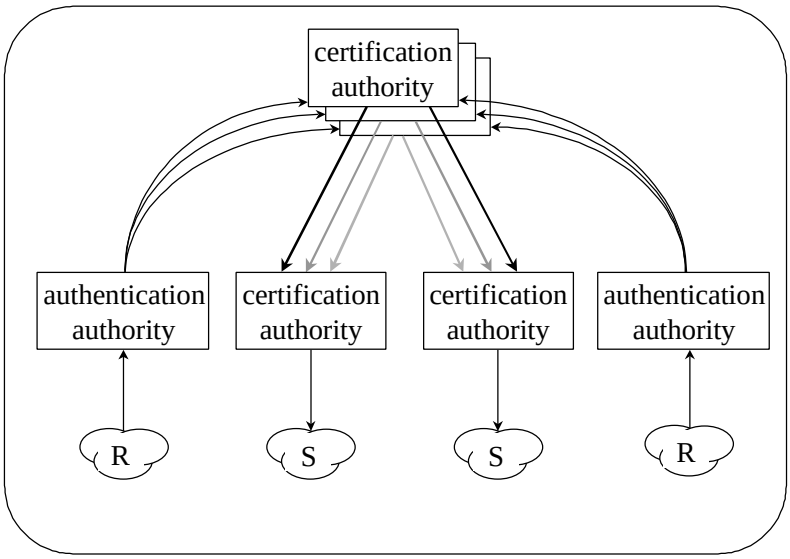

Figure 15 - Tow-tier hierarchy

When the requirement arises to recognize a new root authority, relying parties must import the new root CA's public key. There is currently no standard protocol to support these operations.

Each root CA must guarantee to its relying parties that subordinate CAs and their subscribers implement security policies that are consistent with its declared certificate policy. If all authorities and subscribers are within the boundary of a single legal entity, then direct control may be a practical measure to ensure that this is the case. Otherwise, it must apply less reliable contract provisions.

It is also possible for an authentication authority to allow its relying parties to rely directly on the local subordinate certification authority, bypassing the root CAs for the purposes of local trust decisions, as shown in Figure 16.

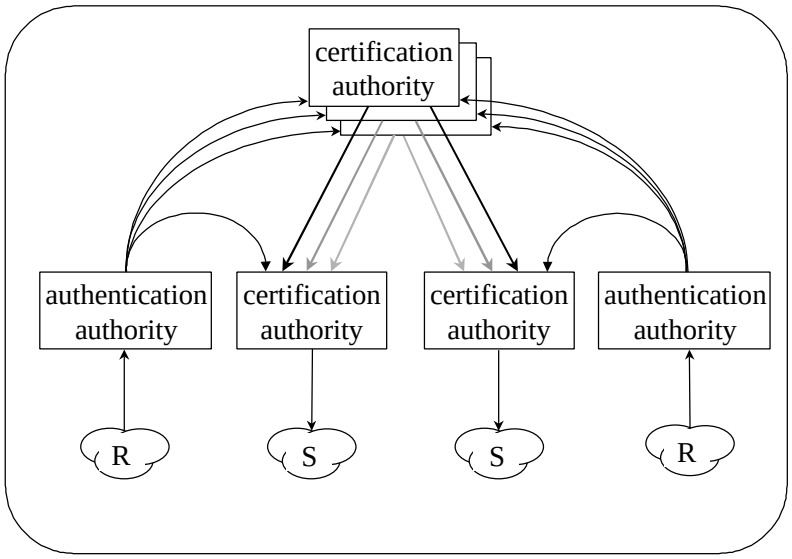

Figure 16 - Two-tier hierarchy with direct local trust

It may be impractical for a certification authority operating subordinate to one root CA to simultaneously subscribe to other root CAs. This is particularly likely where one root CA uses some degree of direct control over the behaviour of its subordinate certification authorities. In this case, the arrangement shown in Figure 17 would be required. However, such an arrangement is unlikely to be acceptable because the cost of operating multiple and inconsistent authorities will be unacceptable. Therefore, for the two-tier hierarchy to be practical, subordinate CAs must be able to operate within multiple root CA certificate policy domains, even when those root authorities are operated by competing root CA service providers, one of whom has direct control over some aspects of the subordinate CA's operation.

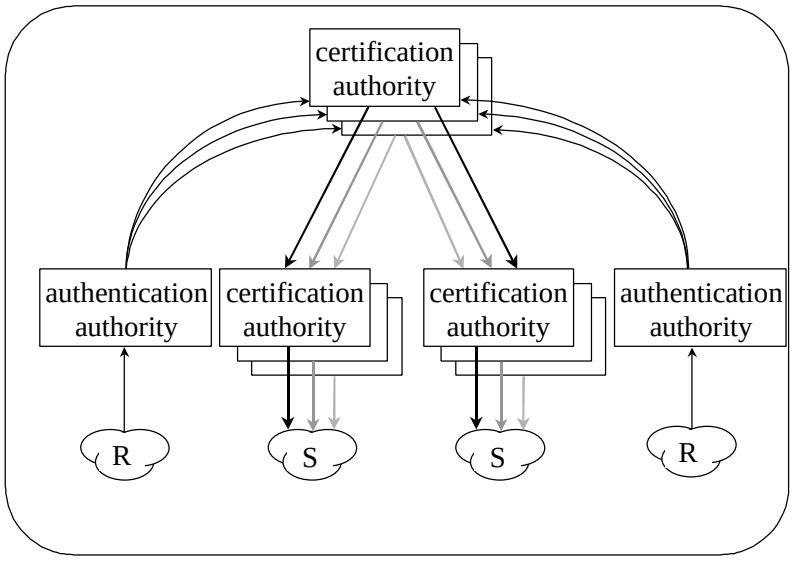

Figure 17 - Independent policy two-tier hierarchies

As the hierarchical trust model is currently implemented in the browser environment, a set of root CA keys is distributed to relying parties as part of the common end-user software. This has the unfortunate effect that CAs do not control the composition of their relying party community and, therefore, the level of risk to which they are exposed. For this reason, this approach may be considered less acceptable in the future.

Hub certification authority

The trust relationships in the hub certification authority architecture are shown in Figure 18. This model architecture is of practical value when a suitable intermediary can be identified.

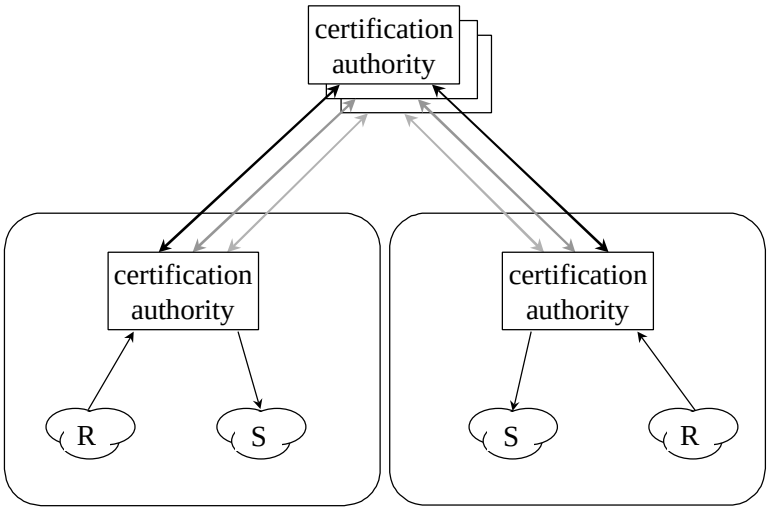

Figure 18 - Hub certification authority

Hub authentication authority

The trust relationships in the "hub authentication authority" trust model are shown in Figure 19. It is similar to the hub cross-certification architecture, except that the authority does not need to operate an automated information system. Authentication tokens can be distributed by procedural means as a precursor to direct cross-certification. However, the authentication authority cannot automatically revoke relationships.

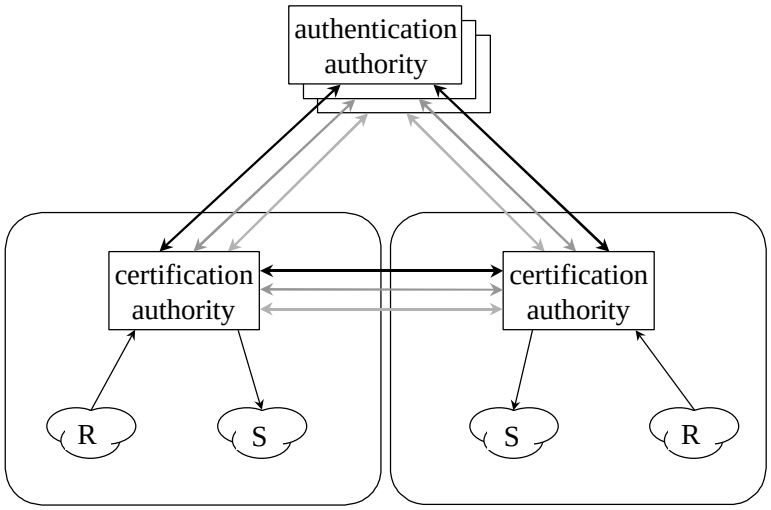

Figure 19 - Hub authentication authority

Network of trust

It has been suggested that a rich network of trust relationship can form a basis for trust management between autonomous organizations. While this is technically feasible, it seems to have no practical value. Contract law requires that all participants have an "interest" in transactions that depend upon them. Therefore, it is not enforceable to rely on parties that are not directly involved in the business transactions flowing between end-users. However, such a network partitioned by policy, such that all valid trust chains are constrained in length and dedicated to a specific purpose, does have practical value.. The use of root and hub certification authorities within an extended network of trust is illustrated in Figure 20.

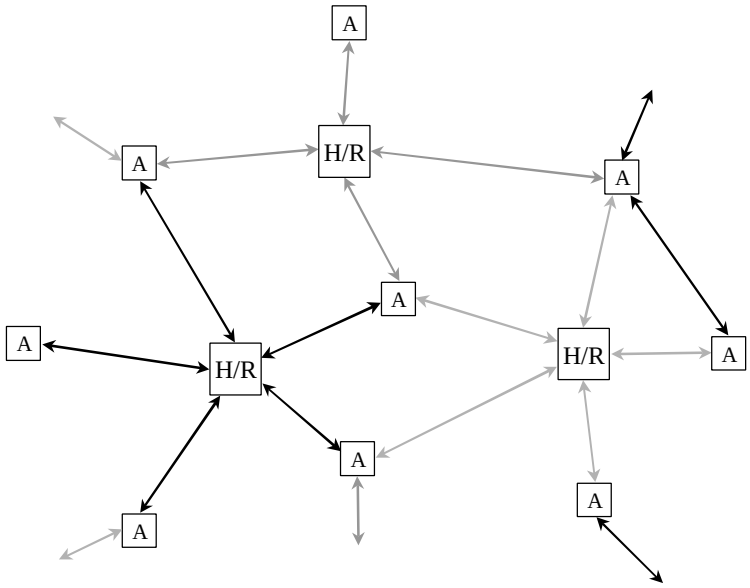

Figure 20 - Network of trust

While it is possible to construct a chain of certificates between any two authorities in the diagram, only those operating under an identical policy will validate according to the standard rules for certificate path validation.

Summary

While, superficially, the hierarchical and hub CA trust models are markedly different, the real difference, when implemented in an organizational setting, stems from the fact that, in the hierarchical model, the relying party relies directly on an authentication authority, rather than a certification authority for its introduction to other parties. So, functional differences in the two approaches flow from the different characteristics of the two authority types, which are laid out above. Primarily, the relying party can be "introduced" to other certification authorities by the certification authorities upon which it relies directly using standard protocols.

In the hierarchical trust model, the relying party specifies its policy requirements by choosing a suitable root key. Whereas, in the distributed trust model, the relying party specifies its policy requirements in the certificate path validation initial policy set.

Related issues

Although not fundamentally linked to the choice of trust model, certain ancillary techniques have become closely associated with that choice. The

advantages and disadvantages of some of these techniques are discussed further here.

Authority public key distribution

The browser practice has brought acceptance to the idea of distributing a set of authority public keys essentially (hard-coded) in the common end-user software. This makes uncertain the extent to which reliance is being placed on an authority. The only reasonable response, on the part of the authority, to this situation is to deny ***all*** liability. Unless the relying party enters into a click-wrap contract in a subsequent step associated with verifying the certificate's status. It would seem to be more satisfactory to qualify the approved uses of keys distributed in this way.

Trust branding

Relying parties need a basis for accepting the assurance of a PKI that a certificate is valid within the certificate policy operated by that PKI. Increasingly, relying parties and subscribers will participate in multiple PKIs. Therefore, it is essential to have a simple, clear and trustworthy mechanism by which to inform relying parties which trust provider they are relying on when they validate a certificate. Therefore, visual trust-branding will be required.

Repository

Internet PKIs have not traditionally incorporated a repository for distributing public-key-related information, because such a mechanism is not supported in the Internet. They have managed to avoid this requirement because of their single-key architecture, their lack of support for revocation and the fact that leaf-certificates can be issued in the form of certificate chains emanating from the root CAs. This approach, however, presumes that new relationships will not be entered into after the leaf-certificate has been issued. The repository allows flexibility to manage relationships with other authorities without reissuing leaf-certificates. Provide support for discovering suitable trust paths, distributing revocation information and discovering public keys in advance of a data exchange (e.g. for data encryption).

Acknowledgments

I am very grateful to Alan Asay for his assistance in preparing this paper.

References

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology, Open systems interconnection - The Directory: authentication framework.

[PKIX] Internet draft certificate policy and certification practice framework.